The Claims

1. (Original) One or more computer-readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a computer, causes the one or more processors to perform the following acts:

receive a request, corresponding to a user, to access a file;

obtain an access control entry that corresponds to both the user and the file, wherein the access control entry includes an encrypted symmetric key that was used to encrypt the file;

check whether a mapping of the access control entry to the symmetric key exists in an encrypted key cache; and

if the mapping exists, then use the mapped symmetric key from the encrypted key cache to decrypt the file, otherwise decrypt the encrypted symmetric key and use the decrypted symmetric key to decrypt the file.

2. (Original) One or more computer-readable media as recited in claim 1, wherein decrypting the encrypted symmetric key comprises using a private key of a public/private key pair associated with the user to decrypt the symmetric key.

3. (Original) One or more computer-readable media as recited in claim 1, wherein the plurality of instructions further cause the one or more processors to perform the following acts:

receive an access control list including a plurality of access control entries;

select one of the plurality of access control entries that corresponds to the user; and

use, as the access control entry, the selected one of the plurality of access control entries.

4. (Original) One or more computer-readable media as rccited in claim 1, wherein the plurality of instructions further cause the one or more processors to perform, if the mapping does not exist, the following:

create, after decrypting the encrypted symmetric key, a new mapping in the encrypted key cache that maps the access control entry to the symmetric key.

5. (Original) One or more computer-readable media as recited in claim 1, wherein the plurality of instructions further cause the one or more processors to perform the following acts:

generate a file including the encrypted key cache;

encrypt the generated file using a private key of a public/private kcy pair associated with the user; and

store the encrypted file.

6. (Original) One or more computer-readable media as recited in claim 1, wherein the plurality of instructions further cause the one or more processors to perform the following acts:

gcnerate a file including the encrypted key cache;

encrypt the generated file with another symmetric key;

generate a new access control entry corresponding to the generated file;

encrypt the other symmetric key with a public key of a public/private key pair associated with the user; and

store both the encrypted other symmetric key and an identifier of the user corresponding to the key cache in the new access control entry.

7. (Original) One or more computer-readable media as recited in claim 1, wherein the plurality of instructions further cause the one or more processors to perform the following acts:

obtain, in encrypted form, the encrypted key cache from a remote storage device;

decrypt the key cache using a private key of a public/private key pair associated with the user; and

use, as the encrypted key cache, the decrypted key cache.

8. (Original) One or more computer-readable media as recited in claim 7, wherein decrypting the key cache comprises:

decrypting, using the private key, a symmetric key corresponding to the key cache; and

decrypting, using the symmetric key corresponding to the key cache, the key cache.

9. (Original) One or more computer-readable media as recited in claim 1, wherein the checking comprises indexing into the encrypted key cache based on the encrypted symmetric key of the access control entry.

10. (Original) One or more computer-readable media as recited in claim 1, wherein the checking comprises indexing into the encrypted key cache based on a user name included in the access control entry.

11. (Original) One or more computer-readable media as recited in claim 1, wherein the plurality of instructions further cause the one or more processors to perform the following acts:

removing one mapping from the encrypted key cache while leaving one or more other mappings in the encrypted key cache.

12. (Original) One or more computer-readable media as recited in claim 11, wherein the removing comprises removing the one mapping if the one mapping has not been accessed within a certain time frame.

13. (Original) One or more computer-readable media as recited in claim 11, wherein the removing comprises removing the one mapping if the encrypted key cache is already full and a new mapping is to be saved in the encrypted key cache.

14. (Original) A method comprising:

receiving an access control entry corresponding to a file and including a symmetric key encrypted with a public key;

checking whether an access control entry to symmetric key mapping exists in a key cache; and

obtaining the symmetric key from the key cache if the mapping exists, otherwise decrypting the encrypted symmetric key using a private key corresponding to the public key.

15. (Original) A method as recited in claim 14, wherein the public key and the private key are both part of a public/private key pair associated with a user.

16. (Original) A method as recited in claim 14, further comprising:

receiving an access control list including a plurality of access control entries;

selecting one of the plurality of access control entries that corresponds to the user; and

using, as the access control entry, the selected one of the plurality of access control entries.

17. (Original) A method as recited in claim 14, wherein if the mapping does not exist, then creating, after decrypting the encrypted symmetric key, a new mapping in the key cache that maps the access control entry to the symmetric key.

18. (Original) A method as recited in claim 14, further comprising:

generating a file including the key cache;

encrypting the generated file using the private key; and

storing the encrypted file.

19. (Original) A method as recited in claim 14, further comprising:

generating a file including the key cache;

encrypting the generated file with another symmetric key;

generating a new access control entry corresponding to the generated file;

encrypting the other symmetric key with the public key; and

storing both the encrypted other symmetric key and an identifier of a user corresponding to the key cache in the new access control entry.

20. (Original) A method as recited in claim 14, further comprising:

obtaining a key cache in encrypted form from a remote storage device;

decrypting the key cache using the private key; and

using, as the key cache, the decrypted key cache.

21. (Original) A method as recited in claim 20, wherein decrypting the key cache comprises:

decrypting, using the private key, a symmetric key corresponding to the key cache; and

decrypting, using the symmetric key corresponding to the key cache, the key cache.

22. (Original) A method as recited in claim 14, wherein the checking comprises indexing into the key cache based on the encrypted symmetric key of the access control entry.

23. (Original) A method as recited in claim 14, wherein the checking comprises indexing into the key cache based on a user name included in the access control entry.

24. (Original) One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 14.

25. (Original) A method comprising:

accessing an encrypted key cache, corresponding to a user, in encrypted form;

obtaining an encrypted symmetric key from an access control entry corresponding to the encrypted key cache;

decrypting the encrypted symmetric key using a private key corresponding to the user;

decrypting the encrypted key cache using the decrypted symmetric key; and

using the encrypted key cache to identify, based on access control entries corresponding to other files, symmetric keys used to encrypt the other files.

26. (Original) A method as recited in claim 25, wherein the using comprises using the private key to decrypt the symmetric key corresponding to another file if the access control entry corresponding to the other file is not included in the encrypted key cache.

27. (Original) A method as recited in claim 26, further comprising storing, in the encrypted key cache, a mapping of the access control entry corresponding to the other file to the decrypted symmetric key.

28. (Original) One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 25.

29. (Original) A system comprising:

a control module to obtain an access control entry corresponding to a file to be accessed by the system, wherein the access control entry includes a symmetric key encrypted with a public key of a public/private key pair;

a key cache to maintain a plurality of mappings each of which maps an access control entry to a symmetric key;

a comparator, communicatively coupled to the control module, to check whether one of the plurality of mappings corresponds to the received access control entry; and

a cryptographic engine, communicatively coupled to the control module, to:

use, if one of the plurality of mappings corresponds to the received access control entry, the symmetric key to which the received access control entry maps to decrypt the file, and

use, if one of the plurality of mappings does not correspond to the received access control entry, the private key of the public/private key pair to decrypt the symmetric key, and then use the decrypted symmetric key to decrypt the file.

30. (Original) A system as recited in claim 29, wherein the system is a computing device in a serverless distributed file system.

31. (Original) A system as recited in claim 29, wherein the system is a computing device in a centralized distributed file system.

32. (Original) A system as recited in claim 29, wherein the control module is further to:

receive an access control list including a plurality of access control entries;

select one of the plurality of access control entries that corresponds to a user of the system; and

use, as the access control entry, the selected one of the plurality of access control entries.

33. (Original) A system as recited in claim 29, wherein the control module is further to create, if one of the plurality of mappings does not correspond to the received access control entry, a new mapping in the key cache that maps the access control entry to the symmetric key.

34. (Original) A system as recited in claim 29, wherein:

the cryptographic engine is further to encrypt, using the private key, another file including the key cache; and

the control module is further to store the encrypted file.

35. (Original) A system as recited in claim 29, wherein:

the cryptographic engine is further to encrypt, using another symmetric key, another file including the key cache, and to encrypt, using the private key, the other symmetric key; and

the control module is further to generate a new access control entry corresponding to the other file, and to store both the encrypted other symmetric key and an identifier of a user corresponding to the key cache in the new access control entry.

36. (Original) A system as recited in claim 29, wherein the control module is further to:

obtain a key cache in encrypted form from a remote storage device;

decrypt the key cache using the private key; and

use, as the key cache, the decrypted key cache.

37. (Original) A method comprising:

accessing an encrypted key cache, corresponding to a user, in encrypted form;

decrypting the encrypted key cache using a private key corresponding to the user; and

using the encrypted key cache to identify, based on access control entries corresponding to other files, symmetric keys used to encrypt the other files.

38. (Original) A method as recited in claim 37, wherein the using comprises using the private key to decrypt the symmetric key corresponding to another file if the access control entry corresponding to the other file is not included in the encrypted key cache.

39. (Original) A method as recited in claim 38, further comprising storing, in the encrypted key cache, a mapping of the access control entry corresponding to the other file to the decrypted symmetric key.

40. (Original) One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 37.

41. (Original) A method as recited in claim 37, further comprising:

rcmoving an entry from the encrypted key cache while leaving one or more other entries in the encrypted key cache.

42. (Original) A method comprising:

accessing a key cache that maintains a plurality of access control entry to symmetric key mappings corresponding to a plurality of files accessible to a user in a distributed file system, wherein each of the plurality of mappings identifies a symmetric key that can be used to decrypt a file corresponding to the mapping;

generating an encrypted file that includes the key cache and that is encrypted using a symmetric key;

encrypting the symmetric key using a public key corresponding to the user;

storing the encrypted symmetric key in an access control entry corresponding to the encrypted file; and

storing both the encrypted file and the access control entry corresponding to the encrypted file in the distributed file system.

43. (Original) One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 42.

44. (Original) A system comprising:

means for storing a plurality of access control entry to symmetric key mappings;

means for retrieving an access control entry corresponding to a requested file;

means for comparing the retrieved access control entry to the plurality of access control entry to symmetric key mappings and for determining whether any of the plurality of mappings match the retrieved access control entry; and

means for obtaining, from the means for storing, a symmetric key to be used to decrypt the requested file if one of the plurality of mappings matches the retrieved access control entry, and otherwise for decrypting the symmetric key, in encrypted form, using a private key of a public/private key pair corresponding to the public key used to encrypt the symmetric key.